N89-15585

AUTOMATED KNOWLEDGE BASE DEVELOPMENT FROM CAD/CAE DATABASES

R Glenn Wright and Mary Blanchard

PROSPECTIVE COMPUTER ANALYSTS, Inc.
1215 Jefferson Davis Highway, Suite 309
Arlington, VA 22202

ABSTRACT

Knowledge base development requires a substantial investment in time, money, and resources in order to capture the knowledge and information necessary for anything other than trivial applications. This paper addresses a means to integrate the design and knowledge base development processes through automated knowledge base development from CAD/CAE databases and files. Benefits of this approach include the development of a more efficient means of knowledge engineering, resulting in the timely creation of large knowledge based systems that are inherently free of error.

INTRODUCTION

Numerous problems traditionally associated with the development of knowledge based systems have been documented, including the availability of experts, the time required to build a system, unfamiliarity of the knowledge engineer with the domain, finding an expert who is enthusiastic about the project, etc.[1][3][10]. Prospective Computer Analysts, Inc, is investigating for NASA's Kennedy Space Center, methods to help resolve these and other problems through automated knowledge base development. Two of the methods and techniques for overcoming these problem areas, automated model building from CAD data and automated knowledge acquisition, are discussed. Each technique is used for generating different types of knowledge.

The automated model-builder generates the part of the knowledge base required for monitoring, control, and diagnosis of a system. The primary advantage associated with this method is a significant reduction in the amount of time and effort required to build a model representative of system connectivity and operational values, both normal and abnormal. Whenever the system design is changed, a new model can be generated quite easily. The knowledge used to generate a model can easily be extended to handle new parts and therefore new designs. Only the routines which directly interface with the CAD files need to be modified for other CAD packages and hardware.

Design knowledge capture techniques, beyond the standard documentation practices traditionally followed, are significantly more difficult to implement. For this particular application, we are referring to capturing design knowledge from the experts. Knowledge will be captured using techniques appropriate for the type of knowledge desired. The design knowledge is captured at the time when it is easiest for the designer to recall: during a design session. In order to diminish the problem of extracting implicit knowledge, indirect knowledge acquisition techniques can also be used.

Using either method, the knowledge is automatically documented and incorporated into the knowledge base in one step. The problem of knowledge verification and validation, however, still remains. This problem is reduced to some degree in the design knowledge capture process, which allow the designer to modify the knowledge base directly to correct any errors produced during it's creation. This capability provides the expert with a control capability over the knowledge base.

AUTOMATED MODEL BUILDING

By using knowledge about classes of components and design data contained in the CAD database, the generation of a model of a system being designed may be automated. This model can be used as part of the knowledge base for monitoring, control, and/or diagnostic software, as well as a communication tool between various people working on the project. For example, designers, test engineers, manufacturing and production personnel could all examine the same design (represented by the model) to check for inconsistencies and other factors throughout the life-cycle of the product. Examples of research in automated model building include that of Thomas [12], and University of Central Florida [6][7].

In order to deal with the vast amounts of information involved, the product being designed may be divided into hierarchial subsystems or modules. Each of these subsystems may be represented as a separate model with connections to the other models, thereby representing the subsystems to which it is connected. Each model would be contained in a separate knowledge base. As each subsystem is needed by the monitoring, control or diagnostic software, this knowledge base can be transferred into memory and the other written back out to disk.

By dealing with files produced from the CAD/CAE database instead of the entire database, the time required to produce a model and the amount of data handling can be reduced substantially. The CAD/CAE files would contain only the design data needed to generate the model for each subsystem, including: a unique name for each component, unique names for all connections between all components, standard nomenclature for each component, units used to measure output flow of a component, range of acceptable values associated with a component, part number (standard or manufacturers) for each component, and the tolerance associated with each measurement. All available measurements, commands, and components within the subsystem, are contained with the files. Information about the direction of flow included in the CAD/CAE database, and in turn the files, would increase the speed at which the model is built. Generally, the CAD/CAE database only indicates how components are physically connected, with no information given about the direction of the signal flow. As VHDL and

EDIF use becomes more widespread, this problem should lessen.

If different versions of a design are allowed, then different models representing these versions of the design must exist. When the designer feels significant changes have been made to a design, a new model must be generated. By requiring the designer to provide meaningful names for the models, including the version number of the design and for any connected subsystems, the model-builder will be able to handle multiple models for the same subsystem. The designer, or design configuration manager, should have the option of erasing models corresponding to old designs. However, the knowledge of the changes which occurred, and why they occurred, will be maintained in the design knowledge base. Neither the designers nor the support group will be allowed to erase the design knowledge base.

The model builder will use knowledge about the type and number of inputs and outputs associated with a component type to derive flow information. A connection list containing components and connection points, will initially be generated from one of the CAD files. The model builder algorithm will look for all the connections between the components listed in the connection list, then determine the input and output connections between the components. Additional information, such as typical or standard names for input and output connections, can be used. If the algorithm is unable to determine the direction of the signal flow for these components, based upon this information, the system will hold off making this decision until more information is known about the other components in the model. It is reasonable to assume that at some point the system will be able to make this decision for one pair of components in the model. Once the decision is made for one set, it narrows down the possibilities for the other components in the model, thus making it possible to determine the directional flow between pairs of components, which were previously eliminated.

Although the benefits of automated model building through this approach are many, certain problems which limit it's utility must be addressed. These problems include when the development of CAD designs is spread out over various, non-compatible, CAD/CAE hardware and software; handling the voluminous amount of information involved; constantly changing designs; the ability for many versions of the same design to exist; and for different designers to be using different versions of the same subsystem within their own design. These problems dictate the standardization of CAD/CAE design environments within common development and/or product lines. This will significantly reduce translation and configuration management requirements, and the resultant errors.

DESIGN KNOWLEDGE CAPTURE

CAD databases maintain the design representation and changes made to the design, however, no method currently exists to capture the reasons behind design decisions and changes. In order to capture this knowledge, it is necessary to supplement CAD/CAE software with a design knowledge capture tool.

By interacting with the designer using voice recognition and voice synthesis, the designer may be interviewed during the evolution of the design. Access to the design is provided through the model created from the CAD/CAE data. At the beginning of each design session, the design knowledge capture tool ensures that it has a model of the subsystem design to be worked on by the designer. If the model does not exist, one can be generated. It would not be practical to continually generate new models during the design session. Using this method, the design knowledge capture tool has an accurate model of the subsystem design at the beginning of the design session, and the designer is asked questions to determine what changes are being made.

Once a change is detected, questions can be asked to capture the designer's knowledge which went into making that change. This method leads the designer into explaining the design planning strategies, related analogies, general design knowledge, and the designers own experiences which went into making the design decisions. The knowledge used by an expert in designing must be represented using different data structures. For example, a plan can be best represented as a cyclic directed graph. The information required for each type of knowledge needs to be explicitly defined in order for questions to be generated. Associated with each question would be a set of expectations which can be compared with the answers received. Information received from the designer may or may not pertain to this question. Extraneous information received from the designer can still be processed by the system and incorporated into the knowledge base, however, the list of expectations will ensure that when an answer is given to the question, it will be recognized.

Typical questions asked could include:

1. (Why are you)(raising : changing : lowering : ...) (the) (pressure : temperature : dimensions : ...) (of the) (compressor : pump : power supply : ...) (?)

2. Why is this change necessary?

3. What other parts will be affected by this change?

4. How will these other parts be affected by this change?

5. Have you seen a similar configuration of parts previously?

Research was performed in learning casual models of physical mechanisms by understanding real-world natural language explanations of these mechanisms by the University of Connecticut [4][11].

Forward chaining rules can be used to select questions based upon the user's responses and related pieces of knowledge in memory. It is therefor very important to establish a relationship between what is being said by the designer and memory. When a human being is reading a sentence, the individual words are recognized, concepts are formed based upon those recognized words, and the user is reminded of related knowledge in memory. This should be the same process which takes place during natural language processing.

Intuitively it would appear a connectionist approach, such as that suggested by Jordan Pollack would be the best implementation for natural language processing [5][9]. This approach provides mechanisms for combining various types of knowledge for the processing of natural language. Each knowledge segment is represented as a node with excited or inhibited links to other nodes within the network. Each of the nodes represents a concept or microfeature within the domain. It allows domain and general world world knowledge and syntactic and semantic constraints to be integrated together for processing of the input.

Analogies, plans, experiences, general design frames, specific design frames, and design rules would be integrated into the network via nodal connections. This permits episodic memory to be used in the processing of the user's input. Nodal connections can be established between concepts or microfeatures within the network and slots of the general design frame for compressors, for example. Once the user mentioned the word "compressor", all of the related plans, experiences, etc. would be activated, in addition to the corresponding concepts and microfeatures within the network.

Some problems associated with this approach exist, however. Of primary concern is that the number of nodes and connections required is enormous. Either the nodes and connections for the entire network need to be represented in memory as they are, or a sub-network will need to be generated as words are processed. The first method will require a significant amount of memory. The second method involves additional overhead and therefore additional time. A parallel processor will be required to generate the new activation values for each of the nodes and update the weighted connections. The connections can be hardwired. Another question is how will the system handle the introduction of new nodes and the establishment of new connections. As each new part is mentioned by the user, it should become a part of the network. It will also be difficult to relate the activated concepts/microfeatures within the network to the changes required in the plans to reflect this new information. It will be difficult to incorporate information received from the designer into the plans, analogies, etc. used to represent the design knowledge.

Another obvious problem is the cost of a parallel processor. It would not be feasible to provide every design engineer with a massively parallel processor to perform natural language processing. An alternative solution is to combine a conceptual analyzer [2] where syntactic and semantic information can be stored, with episodic memory, represented as general design frames, specific design frames, design rules, plans, analogies, and experiences. Episodic memory will be stored in terms of a type of component, a specific component, and/or a characteristic of the component. Activation of a related concept in the dictionary (used in conceptual analysis) would cause the activation of related pieces of episodic memory. This approach will allow syntactic and semantic constraints, domain and general world knowledge with episodic memory to be integrated, although not to the same degree as a connectionist network. Also, the speed of processing will be slower. However, at the present time this method is feasible, while the connectionist approach should be considered appropriate for future applications.

Design Knowledge Capture Considerations

A key element in design knowledge capture is the need to be able to recognize changes in the focus of attention. When the designer changes the course of the conversation, new questions and expectations need to be generated. The currently active knowledge in memory needs to be changed to reflect the new focus of attention. Consideration must be given as to whether to incorporate all of the previous information given prior to the change in focus into the knowledge base, erasing the related questions and expectations, or to keep this information on the chance that the designer will refer to an earlier topic. This will require considerable overhead and the need for questions to be generated to determine which one of the previous subjects the designer is discussing. One possible solution is to consider one subject at a time and look upon any diversion as a change in focus.

Restricting the designers' ability to modify design information given within certain time periods, e.g. daily, weekly, etc., but allowing visibility to previous data is desirable. One possibility is to allow the designer to change knowledge given during the current (i.e., present day) design session, but provide read-only access to knowledge given in prior design sessions. This protects the knowledge of other designers contained within the knowledge base, and prevents the loss of previous knowledge which would be very difficult to replace in the event of accidental loss. Further restrictions with respect to the amount of knowledge which can be modified in the same subject area would also be warranted for the same reason. A designer may also add knowledge to the knowledge base using techniques discussed in the following paragraphs.

Knowledge acquisition techniques fall into one of two categories, direct or indirect [8]. Direct techniques include: interviews, questionnaires, observation of task performance, protocol analysis, interruption analysis, closed curves and inferential flow analysis. Questionnaires can be generated for holes in the knowledge base and the user asked to fill out these questionnaires. Drawing closed curves can be used to help discover analogies. The designer would be asked to draw a closed circle around related objects. Next, questions can be generated to determine the similarities between the objects and derive design rules which can be extended from one domain to another.

The indirect methods include multidimensional scaling, hierarchical clustering, general weighted networks, ordered trees, and repertory grid analysis. General weighted networks can be used to discover planning strategies. The network is made up of concepts represented as nodes. The links are used to show order and direction of the steps in the plan. In this case, the concepts are steps of a plan. The user would be given major steps within a plan and through questions, the substeps of the plan would be discovered. The designer would establish the order of the substeps using links to connect the nodes.

The total memory of the system includes a model of the system designed and the design knowledge which went into the design. However, additional knowledge can be added such as, repair and maintenance data on the parts contained within the system, manufacturing knowledge on how to produce the parts, the list of manufacturing equipment available and their capabilities and limitations, and knowledge about the environment in which the system will be operating. An additional layer can be added which would act as a communication module, permitting people from various departments access to information about the system. By allowing this open exchange of information during the design of the system, the probability of producing a system which can be manufactured and meet operational requirements, the first time, is increased significantly. This knowledge should be available and easily accessible throughout the life cycle of the system. It can be used when a design revision, a new environment for the system, or when a change in manufacturing equipment is being considered.

Other knowledge acquisition modules would need to be developed to deal with these other domains. Also a suitable network of hardware and software would need to be selected. Other factors to consider would be, an increase in security, how often will the knowledge bases be accessed, physical locations of people accessing the information.

CONCLUSIONS

Two methods for model building and design knowledge capture for automated knowledge base development have been presented. Current technology provides the means to address this topic and initiate development with meaningful results which may be applied towards solving many design knowledge capture and knowledge base development problems which exist today.

The means to overcome limitations in today's technology is available, however long term solutions would greatly benefit from connectionist methodologies utilizing massively parallel processing in a standardized CAD/CAE development environment.

REFERENCES

[1] Berry, Diane C., "The Problem of Implicit Knowledge", Expert Systems, August 1987, Vol. 4, No. 3

[2] Gershman, Anatole V., "A Framework for Conceptual Analyzers", Chapter 6, pp 177-197, in Strategies For Natural Language Processing edited by Wendy G. Lehnert and Martin H. Ringle, Lawrence Erlbaum Associates, 1982

[3] Hoffman, Robert R., "The Problem of Extracting the Knowledge of Experts from the Perspective of Experimental Psychology", AI Magazine, Summer 1987

[4] Klimczak, Benjamin "Learning Casual Models of Physical Devices by Understanding Real-World Explanations using a Global Dictionary", Technical Report #86-23, December 1986, University of Connecticut, Storrs, CT 06268

[5] Lehnert, Wendy G., "Knowledge-Based Natural Language Understanding: A AAAI-87 Survey Talk", COINS Technical Report 88-02, University of Massachusetts, Amhert, Massachusetts

[6] Dr. Myler, H.R. and Dr. Gonzalez, A.J., "NASA Research Grant, Automated Knowledge Generation First Quarterly Review Meeting", Computer Engineering Department, University of Central Florida, Orlando, Florida

[7] Dr. Myler, H.R. and Dr. Gonzales, A.J., "NASA Research Grant, Automated Knowledge Generation, Second Quarterly Review Meeting", Computer Engineering Department, University of Central Florida, Orlando, Florida

[8] Olson, Judith Reitman and Rueter, Henry H. "Extracting Expertise from Experts: Methods for Knowledge Acquisition, Expert Systems, August 1987, Vol. 4, No. 3

[9] Pollack, Jordan B., "On Connectionist Models of Natural Language Processing", MCCS-87-100, New Mexico State University, Las Cruces, New Mexico

[10] Prerau, David S., "Knowledge Acquisition in the Development of a Large Expert System", AI Magazine, Summer 1987

[11] Selfridge, Mallory and Cuthill, Barbara, "Automatic Invention: Inferring the Casual Model of a Physical Mechanism From a Description of Desired Behavior", Technical Report #86-17, December 1986, University of Connecticut, Storrs, CT 06268

[12] Thomas, Stan J. "NASA/ASEE Summer Faculty Research Fellowship Program, Automated Construction of a Knowledge Base From Computer Aided Design Data", Wake Forest University